Informe de Lenguajes, Paradigmas y Estándares de Programación

- Introducción

Los lenguajes de programación hoy en día son fundamentales en el mundo tecnológico. Esta herramienta permite la creación de software, aplicaciones… Para dichas creaciones esta herramienta da lugar a una serie de valores los cuales posteriormente son transformados en softwares. Los lenguajes permiten que los humanos creen una serie de instrucciones a una máquina permitiendo la automatización de actividades. Las ofertas laborales que te habilita la programación son enormes. Es algo reciente y complicado que hace unos años no se veía como importante pero hoy en día la demanda es muy alta.

Los paradigmas de la programación facilitan la construcción de códigos legibles y organizados. Además, hay diferentes tipos de paradigmas los cuales son posteriormente utilizados en diferentes situaciones. Los diferentes paradigmas para cada situación facilitan una productividad y una eficiencia mayor ya que sus herramientas se adaptan al trabajo. Cada tipo tiene su estructura que es universal, así que la comunicación entre equipos de trabajo es mucho más fácil.

Los estándares de programación son los comandos básicos que ese utilizan en los lenguajes de programación. Son una serie de reglas que determinan como debe escribirse el código. Los estándares facilitan la lectura del código para el humano ya que a la máquina le da igual mientras funcione.

- Tipos de lenguaje de programación

Los lenguajes de programación de **alto nivel** no expresan algoritmos teniendo en cuenta las capacidades de ejecución de la máquina. El alto nivel utiliza un lenguaje más cercano al humano y necesita luego un intérprete para traducir el código. Su objetivo es ir más allá sobre las limitaciones de los lenguajes de bajo nivel, permitiendo a los usuarios resolver problemas fácilmente. Ejemplos de este tipo de lenguajes están: Python, Java, C++, JavaScript... Según el propósito del lenguaje, hay diferentes tipos.

Los lenguajes de *propósito general* permiten crear programas y algoritmos de todo tipo sin especificarse en una sola área.

Propósito específico, son lenguajes que se dedican a trabajar en tareas específicas y cuentan con características particulares y mucha abstracción.

Según su método de ejecución pueden ser lenguajes compilados o lenguajes interpretados.

Los lenguajes de programación de **medio nivel** se denominan así ya que normalmente son clasificados de alto nivel, pero permiten ciertos manejos de los lenguajes de bajo nivel. Su uso más común y principal es para la creación de sistemas operativos los cuales necesitan un manejo abstracto, pero también el poder y la eficiencia de los lenguajes de bajo nivel. Este tipo de lenguaje te permite tener un equilibrio entre la facilidad de la programación y el control sobre el hardware. Lenguajes utilizados en este nivel son: C++, C…

Los lenguajes de programación de **bajo nivel** utilizan un lenguaje más cercano al de la máquina y ofrece un control sobre el hardware. El bajo nivel no es ningún lenguaje en concreto, si no que engloba varios tipos de lenguajes. Dentro de este nivel existen tres tipos.

El *código binario* es la base que existe dentro de todos los sistemas informáticos. Es sencillo de utilizar compuesto por solo dos caracteres (0,1).

El *lenguaje de máquina* está también formado por 0 y 1, pero este lenguaje manda instrucciones directamente a la CPU.

El *lenguaje ensamblador* es un lenguaje abstracto el cual necesita una herramienta para traducir el código para que la CPU pueda ejecutarlo, pero estos programas no son ejecutados directamente por el ordenador.

- Tipos de paradigmas de la programación

El **paradigma imperativo**, es el paradigma más antiguo, consta de una secuencia definida de instrucciones de un ordenador. El código encadena diferentes instrucciones una tras otra determinando lo que debe hacer el ordenador en cada momento. Los valores utilizados cambian durante la

ejecución de las instrucciones y para gestionar dichas instrucciones se integran estructuras de control. Dentro del paradigma imperativo se encuentran tres estilos distintos.

El *estilo estructurado* es un tipo donde el flujo de control se define mediante bucles, condicionales y subrutinas. Amplía también el principio imperativo central con las estructuras concretas de control. Esto evita las instrucciones que añaden complejidad innecesaria. Como conclusión, busca imponer restricciones a la transferencia directa de control, con el propósito de establecer una estructura más flexible. Algunos ejemplos de lenguajes que son utilizados en este paradigma son C++, C, COBOL, FORTRAN, Pascal, etc.

El *estilo procedimental* divide las tareas de las que se debe ocupar un programa en tareas más pequeñas. De esta forma se consigue que el código sea más claro y no haya repeticiones innecesarias en el código gracias a las llamadas de las funciones y procedimientos. Lenguajes que utilizan este estilo como Python y C#.

El *estilo modular* muestra que cada uno de los componentes del programa se diseñan, desarrollan y prueban con independencia del resto. El software no es finalizado hasta que todos los componentes están listos y en ese momento se combinan.

El *estilo orientado a objetos,* todos los elementos son tratados como un objeto con sus propios atributos. Este estilo es muy sencillo ya que si se necesita una función se añade otro objeto. Las clases que cada objeto tiene pueden ser heredadas por una clase superior o superclase. Este estilo va a ser utilizado en programas grandes que tienen que ser actualizados regularmente. Ejemplo de lenguajes que utilizan este estilo como Java o Ruby.

El **paradigma declarativo**, lleva a cabo una descripción del resultado final, en lugar de mostrar los pasos. Se determina directamente la vía de solución. Este método funciona cuando las especificaciones del resultado son claras y también que exista un procedimiento adecuado. Cuando estas dos condiciones están presentes, este paradigma es muy efectivo. Al no determinar los pasos para

obtener dicho resultado, da paso a la optimización. Dentro del paradigma declarativo se encuentran otros tipos.

La *programación funcional* crea programas utilizando matemáticas. Esto es utilizado en lenguajes como Haskell y Lisp. Esta programación se enfoca en la definición y la composición de las funciones para expresar la lógica del programa.

La *programación lógica*, permite la expresión de relaciones y reglas sobre un dominio de problema. Los desarrolladores que utilizan lenguajes de programación lógica especifican un conjunto de axiomas, hechos y reglas, y el motor de razonamiento del lenguaje deriva conclusiones y respuestas. Haskell es un ejemplo de un lenguaje el cual utiliza esta programación y también utiliza como se ha comentado anteriormente el funcional.

La *programación de flujo de datos* enfatiza el flujo de datos a través de una red de procesos o funciones. Los programas suelen estar compuestos por componentes que transforma o filtran datos que fluyen entre ellos. Lenguajes como LabVIEW y Pure Data utilizan esta programación. Utilizándose a menudo en el procesamiento de señales, la simulación y la programación basada en imágenes.

La *programación basada en restricciones* es otro enfoque donde los desarrolladores definen variables, constantes y relaciones y restricciones entre dichos elementos. Esta programación implica la búsqueda de soluciones. Se ve en lenguajes como ECLiPSe y Mozart/Oz.

- Los estándares de la programación

Los estándares de programación son las pautas que determinan la forma de codificar programas según el lenguaje utilizado. Nos determinará como representar sus variables, estructuras, tablas, etc. La importancia del uso de los estándares se debe a que hay que asegurarse de que el desarrollo de dicha codificación se puede ver bien en la actualidad y en el futuro. Los estándares ayudarán a que todas las personas que contribuyan en un proyecto hayan utilizado una única forma de diseñar el código y esto da resultado a que todo tenga sentido y sea fácil de leer para todo el mundo. Su uso es importante

también ya que ayuda a que el software sea de calidad, pero tener esta calidad no es tan fácil porque requiere mucho trabajo y constancia.

Por ejemplo, existe el estándar del estilo de código, el cual define reglas que hay que seguir sobre la apariencia y formato del código. Esto incluye el espacio entre líneas, etc.

El estándar de codificación segura significa que los desarrolladores aplican un conjunto de estándares que ayuda a prevenir y acabar con vulnerabilidades como ataques cibernéticos. Este estándar es la primera línea de defensa contra los ataques y explotaciones del software.

El estándar de nomenclatura es esencial en el mundo de la programación porque ayuda a escribir un código que tenga una facilidad comprensiva, que esté ordenado y sea de calidad. Este estándar tiene un número de reglas las cuales hay que seguir y determinará las variables, tipos, funciones y el resto de las entidades que se utilizarán en el código. Esto ayuda a que cualquier persona sea capaz de leer el código.

Los estándares ofrecen más ventajas que desventajas. Por ejemplo, varias ventajas que ofrecen son la detección temprana de fallos que pueda tener el código. Reduce la complejidad y es más fácil de navegar por el código además de que lo hace reutilizable.

Si no se utilizasen dichos estándares, no tendremos lo que nos ofrece y eso causaría numerosos problemas además de que luego sería todo más difícil de utilizar y de mantener.

- Conclusión

Con la finalización de este trabajo he llegado a aprender y a entender las numerosas reglas que tiene el mundo de la programación. Cada elemento en la programación tiene su sitio y su función y muchos de ellos pueden trabajar juntos.

- Referencias

https://www.ucatalunya.edu.co/blog/los-mejores-lenguajes-de-programacion-en-el-desarrollo-web-y-su-importancia-en-la-era-digital#:~:text=El%20lenguaje%20de%20programación%20es,ordenador%2C%20que%20luego%20lo%20ejecuta.

https://www.etac.edu.mx/blog-etac/index.php/lenguajes-de-programacion-2#:~:text=Precisamente%2C%20los%20lenguajes%20de%20programación,modernización%20general%20de%20las%20organizaciones.

https://www.campusmvp.es/recursos/post/las-principales-ventajas-de-aprender-a-programar-en-varios-lenguajes-o-plataformas.aspx#:~:text=Los%20lenguajes%20de%20programación%20te,herramienta%20para%20un%20proyecto%20determinado.

CHATGPT

https://profile.es/blog/que-son-los-paradigmas-de-programacion/#:~:text=Un%20paradigma%20de%20programación%20es,resultados%20que%20necesitan%20los%20programadores.

https://beecrowd.io/es/blog/paradigmas-de-programacion/#:~:text=LA%20IMPORTANCIA%20DE%20APRENDER%20SOBRE,la%20productividad%20diaria%20del%20desarrollador.

CHATGPT

https://www.tusclasesparticulares.com/questions/programacion/que-es-un-estandar-de-lenguaje-de-programacion#:~:text=Un%20estándar%20en%20lenguaje%20de,para%20todos%20los%20demás%20lenguajes.

https://academy.leewayweb.com/como-garantizar-estandar-codificacion-php/#:~:text=Un%20estándar%20de%20codificación%20es,todo%20lo%20demás%20da%20igual).

https://es.parasoft.com/blog/an-ounce-of-prevention-software-safety-security-through-coding-standards/#:~:text=Los%20estándares%20de%20codificación%2C%20que,alta%20calidad%2C%20seguridad%20y%20protección.

CHATGPT

https://www.universidadviu.com/es/actualidad/nuestros-expertos/lenguaje-de-alto-nivel-los-mas-utilizados

https://miformacion.eu/blog/lenguaje-de-alto-nivel-programacion/

https://conceptobasicodecomputacion.weebly.com/lenguaje-de-alto-medio-y-bajo-nivel.html

CHATGPT

https://conceptobasicodecomputacion.weebly.com/lenguaje-de-alto-medio-y-bajo-nivel.html
https://www.epitech-it.es/lenguaje-bajo -

nivel/#:~:text=Un%20lenguaje%20de%20programación%20de,las%20computadoras%20que%20lo%20soportan.

CHATGPT

https://profile.es/blog/que-son-los-paradigmas-de-programacion/

https://keepcoding.io/blog/paradigmas-de-programacion/

https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/programacion-imperativa/

https://www.hostingplus.com.es/blog/paradigmas-de-programacion-que-son-y-sus-diferentes-tipos/?gclid=EAIaIQobChMI1MS_-PGC9wIV_WpvBB2bKQbnEAAYASAAEgIJQvD_BwE

https://universidadeuropea.com/blog/programacion-orientada-objetos/#:~:text=La%20programación%20orientada%20a%20objetos%20es%20un%20modelo%20de%20programación,lógica%20necesaria%20para%20esa%20manipulación.

CHATGPT

https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/programacion-declarativa/

https://appmaster.io/es/glossary/programacion-declarativa

https://prezi.com/p/k6wgs_qh9n72/los-estandares-de-programacion/

https://blog.thedojo.mx/2021/10/05/estandares-de-calidad-en-el-software.html

https://es.parasoft.com/blog/secure-coding-standards-enforcing-secure-coding-practices-with-sast/#:~:text=Los%20estándares%20de%20codificación%20segura,mantener%20estándares%20de%20codificación%20seguros.

https://aulab.es/noticia/158/convenciones-de-nomenclatura-en-la-programacion-informatica

CHATGPT